



Extensions of the Prototype Application Protocol of Ready-to-Wear Apparel Pattern Making

Y. Tina Lee

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Manufacturing Systems Integration Division
Gaithersburg, MD 20899

QC
100
.U56
NO.5727
1995

NIST

Extensions of the Prototype Application Protocol of Ready-to-Wear Apparel Pattern Making

Y. Tina Lee

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Manufacturing Systems Integration Division
Gaithersburg, MD 20899

October 1995



U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

Extensions of the Prototype Application Protocol of Ready-to-Wear Apparel Pattern Making

by

Y. Tina Lee

National Institute of Standards and Technology

October, 1995

ABSTRACT

A prototype application protocol for ready-to-wear pattern making was developed for the communication of pattern pieces and related information between various CAD systems. The basic methodology for the application protocol is the Standard for the Exchange of Product Model Data (STEP). STEP is an emerging international standard, the result of an effort to develop a mechanism for digitally representing the physical and functional characteristics of a product throughout the product's life cycle. The STEP application protocol addresses a specific application area and contains an information model written in the conceptual modeling language *EXPRESS*.

The experience gained through the implementation of the prototype application protocol and recommendations received from apparel researchers provided us useful input to improve the prototype application protocol. This report presents a revised version of the information model using both the *EXPRESS* language and the *EXPRESS-G* graphical notation. An application activity model that defines the scope and information requirements of the prototype application protocol is also presented in this report. Neither the *EXPRESS-G* diagrams nor the application activity model was shown in the original prototype application protocol report.

Keywords: activity model, apparel, application protocol, APDES, data exchange, *EXPRESS*, information model, pattern, product data, STEP

1 INTRODUCTION

The apparel industry has used computer systems to automate many of its manufacturing processes. However, the manufacturing innovations often stand alone as "islands of automation." Integrating the separate automated processes could greatly improve the effectiveness of the entire enterprise. In recent years, the National Institute of Standards and Technology (NIST), under the sponsorship of the Defense Logistics Agency (DLA), has been developing the apparel product data exchange standard (APDES) [1]. The goal of the APDES project is to develop a comprehensive specification for sharing apparel product data among all stages of the product life cycle.

The APDES project team at NIST, has determined a preliminary set of manufacturing data

interfaces that could be standardized for the effective integration of the information systems required to operate an apparel manufacturing enterprise [2]. The specification of these manufacturing data interfaces is being developed based on the Standard for the Exchange of Product Model Data (STEP) methodology [3]. STEP includes definitions of information models and mechanisms for representing the models and related data. The information models provide a mechanism to communicate the structure and the semantics of the data necessary to exchange among different computer systems and environments. A STEP application protocol (AP) addresses a specific application area and includes the definitions of scope, context, and an information model of the application area. The APDES project team has developed the first prototype application protocol, ready-to-wear pattern making, which is one of the manufacturing data interfaces being identified [4]. The prototype AP specifies the information necessary to represent two-dimensional flat patterns for the purpose of facilitating communication between apparel CAD/CAM ready-to-wear pattern making systems. The information model for the prototype AP was written in the *EXPRESS* language [5,6]. A prototype AP is not developed in full conformance with STEP standards, and it is not integrated with existing STEP resource models. The prototype AP will serve as a straw-man AP to help in the development of formal APs for the apparel industry.

Through working with pattern makers, software developers and apparel researchers, we have received comments/recommendations from them. Section 2 of this report presents a new version of information model of ready-to-wear pattern making revised based on the comments/recommendations received. Additions or modifications to the original information model provide additional features to accommodate information requirements for various apparel CAD systems. To aid readability and visibility, the information model is also shown in *EXPRESS-G* in section 3. *EXPRESS-G* is a formal graphical notation for the display of data specifications defined in the *EXPRESS* language. Section 4 of this report presents an application activity model of apparel production processes to clarify the context and scope of the application protocol of ready-to-wear pattern making. The application activity model is documented using the IDEF0 methodology (refer to "Federal Information Processing Standards Publication 183, Integration Definition for Function Modeling: IDEF0", FIPS PUB 183, National Institute of Standards and Technology, Draft, December 1993.) Use of the information model is discussed in the last section of this report.

2 INFORMATION MODEL

In this section an *EXPRESS* information model for ready-to-wear pattern making (RWPM) is presented. Appendix A contains the listing of *EXPRESS* keywords that are used in the RWPM schema. The information model defines the data types that can be used to define two-dimensional patterns generated by the traditional ready-to-wear pattern making and grading. An *EXPRESS* schema is composed of type declarations, entities, and constraints. Section 2.1 lists the types and entities, with the associated section numbers from this documentation. The RWPM schema is presented in detail in section 2.2.

2.1 List of Types and Entities

TYPES:

composite_curve_feature_type	(Section 2.2.3)
mark_feature_type	(Section 2.2.1)
measurement_unit_type	(Section 2.2.4)
orientation_constraint_type	(Section 2.2.2)
pattern_mirror_type	(Section 2.2.5)

ENTITIES:

annotation_feature	(Section 2.3.10)
arc	(Section 2.3.12)
basic_pattern_piece	(Section 2.3.15)
bounded_curve	(Section 2.3.11)
composite_curve_feature	(Section 2.3.14)
grade_data_at_point	(Section 2.3.21)
grade_delta	(Section 2.3.19)
grade_point	(Section 2.3.18)
grade_rule_at_point	(Section 2.3.20)
grade_rules_of_piece	(Section 2.3.23)
grade_rules_of_pattern	(Section 2.3.24)
library_rule_at_point	(Section 2.3.22)
line	(Section 2.3.3)
mark_feature	(Section 2.3.5)
notch_feature	(Section 2.3.6)
orientation_constraint	(Section 2.3.9)
pattern	(Section 2.3.17)
pattern_geometry_entity	(Section 2.3.4)
pattern_piece	(Section 2.3.16)

pattern_size	(Section 2.3.1)
point	(Section 2.3.2)
polyline	(Section 2.3.13)
slit_notch	(Section 2.3.8)
ready_to_wear_pattern	(Section 2.3.25)
v_notch	(Section 2.3.7)

The entities include a broad range of data types, from simple points to complex entities such as patterns. The way these entity classes are related is specified by the model schema, which is described next.

2.2 RWPM Schema

This section describes the detailed information for the RWPM schema. Types and entities are defined formally here. The schema is presented in a “bottom-up” order: type definitions are presented first, followed by entity definitions. The more specific entity models are described before they are used in the definition of more complex entities. In *Express* language, a “remark” is used for documentation and is not significant as a language element. The character pair, “(” and “*”, is used to denote the start of an embedded remark, and the character pair, “*” and “)”, is used to denote its end. An embedded remark may appear between any two tokens. In this report, the documentation is presented as embedded remarks. Consequently, this entire report can be read into an *Express* parser for further analysis.

*)

SCHEMA rwpm_schema;

(*

2.2.1 Type Definitions

This section contains the type definitions.

2.2.1.1 Mark Feature Type

A *mark_feature_type* provides the means to indicate a mark on the pattern. This is used as an aid for subsequent cutting and sewing processes. *Mark_feature_type* is an enumeration of drill hole, lift and plunge point, stacking point, facing point, and cut entry point. It is used as the type of an attribute defined in the *mark_feature* entity.

*)

TYPE mark_feature_type = ENUMERATION OF

```

(drill_hole,
lift_and_plunge_point,
stacking_point,
facing_point,
cut_entry_point);

END_TYPE;

(*)

```

2.2.1.2 Orientation Constraint Type

An *orientation_constraint_type* provides a means of specifying the orientation of the pattern piece or supporting symmetrical pattern pieces. The pattern piece may be aligned with the fabric's grain, or with some feature of the fabric's decorative design. *Orientation_constraint_type* is an enumeration of grain reference line, stripe reference line, plaid reference line, and mirror line. It is used as the type of an attribute defined in the *orientation_constraint* entity.

```

*)

TYPE orientation_constraint_type = ENUMERATION OF

    (grain_reference_line,
    stripe_reference_line,
    plaid_reference_line,
    mirror_line);

END_TYPE;

(*)

```

2.2.1.3 Composite Curve Feature Type

A *composite_curve_feature_type* provides a means of expressing the purpose of a curve on a pattern piece. This type is an enumeration of boundary cut, internal cut out, fold line, and sew line. It is used as the type of an attribute defined in the *composite_curve_feature* entity.

```

*)

TYPE composite_curve_feature_type = ENUMERATION OF

    (boundary_cut,

```

```

        internal_cut_out,

        fold_line,

        sew_line);

END_TYPE;

(*)

```

2.2.1.4 Measurement Unit Type

A *measurement_unit_type* provides a means to define the units of length. The units of length refers to the basic units of measurement employed for the given model data. *Measurement_unit_type* is an enumeration of centimeter and inch.

```

*)

TYPE measurement_unit_type = ENUMERATION OF

    (centimeter,

    inch);

END_TYPE;

(*)

```

2.2.1.5 Pattern Mirror Type

A *pattern_mirror_type* provides the means to identify the mirror information of the pattern piece. It is an enumeration of basic, horizontal mirror, and vertical mirror. A mirror pattern is a mirror-image of the basic pattern in the horizontal or vertical direction. *Pattern_mirror_type* is used as the type of an attribute defined in the *pattern_piece* entity.

```

*)

TYPE pattern_mirror_type = ENUMERATION OF

    (basic,

    horizontal_mirror,

    vertical_mirror);

END_TYPE;

(*)

```

2.2.2 Entity Definitions

This section defines the entities for the RWPM schema. All pattern geometry is defined in a Cartesian coordinate system.

2.2.2.1 Pattern Size

A garment size designation is an arbitrary name or number for a given compilation of anthropometric measurements. With this designation, the garment will fit someone whose measurements lie within certain range limits of the size measurements. A *pattern_size* entity is represented in terms of an abstract size number (a real number) and/or, an alternate size (a text string). Size number may be the same as an actual body dimension (e.g., neck size for men's shirts) or maybe a number that is not the same as the body measurements used to establish the size (e.g., women's dress size). Alternate size may be used to define a size group or to define some body dimensions that are important to the type of garment but not specified in the size number. Size group is used to specify a group which may be based on length (e.g., short, regular, or long), or may be based on proportion (e.g., Junior, Miss, or Woman). Body dimensions may include circumferential designation and longitudinal designation.

```
*)  
  
ENTITY pattern_size;  
  
    size_number: OPTIONAL REAL;  
  
    alternate_size: OPTIONAL STRING;  
  
WHERE  
  
    at_least_one: EXISTS (size_number) OR  
  
                EXISTS (alternate_size);  
  
END_ENTITY;  
  
(*
```

2.2.2.2 Point

A *point* entity specifies a location on a pattern piece. It consists of an *X* value and a *Y* value. Coordinates are defined from an unspecified origin, determined by the application.

```
*)  
  
ENTITY point;  
  
    x, y: REAL;
```

END_ENTITY;

(*

2.2.2.3 Line

A *line* entity defines a line segment. It consists of two *points*.

*)

ENTITY line;

location1, location2: point;

END_ENTITY;

(*

2.2.2.4 Pattern Geometry Entity

A *pattern_geometry_entity* is an abstraction of *mark_feature* entity, *notch_feature* entity, *orientation_constraint* entity, *annotation_feature* entity, and *composite_curve_feature* entity. (Note: The abstract supertype entity is defined for classification purposes only, and is not ever directly instantiated.)

*)

ENTITY pattern_geometry_entity

ABSTRACT SUPERTYPE OF (

ONEOF (mark_feature,

notch_feature,

orientation_constraint,

annotation_feature,

composite_curve_feature));

END_ENTITY;

(*

2.2.2.5 Mark Feature

A *mark_feature* entity, a subtype of *pattern_geometry_entity*, specifies a drill hole, a lift and plunge point, a stacking point, a facing point, or a cut entry point on the pattern piece. It is a *point*

together with a *mark_feature_type*.

*)

ENTITY mark_feature

SUBTYPE OF (pattern_geometry_entity);

feature_type: mark_feature_type;

location: point;

END_ENTITY;

(*

2.2.2.6 Notch Feature

A *notch_feature* entity, a subtype of *pattern_geometry_entity*, specifies a notch on the seam line or on the perimeter of a pattern piece. Notches are represented in a variety of ways. A notch base point may be given with angle and depth information to define an angled slit notch. Alternately, a v-notch is defined by the notch base point with depth and width. Thus a *notch_feature* is an abstraction of *v_notch* entity and *slit_notch* entity. Both subtypes inherit the base point and the base line that the notch is associated with.

*)

ENTITY notch_feature

ABSTRACT SUPERTYPE OF (

ONEOF (v_notch,

slit_notch))

SUBTYPE OF (pattern_geometry_entity);

notch_base_point: point;

notch_base_line: OPTIONAL bounded_curve;

END_ENTITY;

(*

2.2.2.7 V Notch

A *v_notch* entity, a subtype of *notch_feature* entity, specifies a v-shaped cut on the pattern piece. It is defined by depth and width of the notch.

```

*)

ENTITY v_notch

    SUBTYPE OF (notch_feature);

    depth, width: REAL;

END_ENTITY;

(*)

```

2.2.2.8 Slit Notch

A *slit_notch* entity, a subtype of *notch_feature* entity, specifies an angled slit cut on the pattern piece. It is defined by a depth value and an angle of a notch. The angle specifies an angular dimension, measured in degrees.

```

*)

ENTITY slit_notch

    SUBTYPE OF (notch_feature);

    depth, angle: REAL;

END_ENTITY;

(*)

```

2.2.2.9 Orientation Constraint

An *orientation_constraint* entity, a subtype of *pattern_geometry_entity*, is a direction specification on the pattern piece. It is characterized by a *line*, that defines the location of the orientation on the drawing, together with an *orientation_constraint_type*.

```

*)

ENTITY orientation_constraint

    SUBTYPE OF (pattern_geometry_entity);

    feature_type: orientation_constraint_type;

    location: line;

END_ENTITY;

(*)

```

2.2.2.10 Annotation Feature

An *annotation_feature* entity, a subtype of *pattern_geometry_entity*, is an annotation feature on the pattern piece presented for informational purposes; it does not represent a feature of the cut piece. It consists of a text string and a *line* for locating and orienting text. The *annotation_feature* entity may be used to define style lines, user defined internal lines, etc.

*)

ENTITY *annotation_feature*

SUBTYPE OF (*pattern_geometry_entity*);

text: STRING;

location: line;

END_ENTITY;

(*

2.2.2.11 Bounded Curve

A *bounded_curve* entity is an abstraction of *arc* entity and *polyline* entity.

*)

ENTITY *bounded_curve*

ABSTRACT SUPERTYPE OF (

ONEOF (*arc*,

polyline));

END_ENTITY;

(*

2.2.2.12 Arc

An *arc* entity, a subtype of *bounded_curve* entity, specifies an arch-shaped segment of a curve for which no particular geometric form, such as circle, ellipse, or parabola, is specified. It consists of three *points*: a start point, an intermediate point, and an end point.

*)

ENTITY *arc*

```

SUBTYPE OF (bounded_curve);

start_point, intermediate_point, end_point: point;

END_ENTITY;

(*)

```

2.2.2.13 Polyline

A *polyline* entity, a subtype of *bounded_curve* entity, is a bounded curve of line segments. It consists of an ordered collection of *points* that are connected using straight lines.

```

*)

ENTITY polyline

SUBTYPE OF (bounded_curve);

points: LIST [2:?] of point;

END_ENTITY;

(*)

```

2.2.2.14 Composite Curve Feature

A *composite_curve_feature* entity, a subtype of *pattern_geometry_entity*, is a geometry entity that defines a curve in the drawing. It is a composite curve (an ordered collection of *bounded_curves* joined end to end), together with a *composite_curve_feature_type*.

```

*)

ENTITY composite_curve_feature

SUBTYPE OF (pattern_geometry_entity);

feature_type: composite_curve_feature_type;

composite_curve: LIST [1:?] of UNIQUE bounded_curve;

END_ENTITY;

(*)

```

2.2.2.15 Basic Pattern Piece

A *basic_pattern_piece* entity defines the base shape of one pattern piece of a garment for a particular size. It has identification information, a set of *pattern_geometry_entities*, and

optionally, a tolerance value (a real number). The identification information includes a piece name and optionally, a description. The piece name is a unique pattern piece name within a pattern. The description is a piece identification for the operator to use. If a tolerance value is defined in a *pattern* entity instance, the value is applied to all pattern pieces of the garment. However, the *basic_pattern_piece* allows this value to be redefined. Thus if a tolerance value is defined in a *basic_pattern_piece* entity instance, it overrides the tolerance value that is given at the *pattern* entity instance. A *basic_pattern_piece* may be repeated either as is, or in a mirrored form, for one garment (e.g. identical, but mirrored, pieces for shirt sleeves).

*)

ENTITY basic_pattern_piece;

piece_name: STRING;

description: OPTIONAL STRING;

geometry_entities: LIST [1:?] OF UNIQUE pattern_geometry_entity;

tolerance: OPTIONAL REAL;

UNIQUE piece_name;

END_ENTITY;

(*

2.2.2.16 Pattern Piece

A *pattern_piece* entity defines a pattern piece of a garment for a particular size and the total number of this piece required in the garment. It is defined by a *basic_pattern_piece* with a *pattern_mirror_type*, and a quantity of the pattern piece.

*)

ENTITY pattern_piece;

piece: basic_pattern_piece;

mirror_type: pattern_mirror_type;

quantity: INTEGER;

END_ENTITY;

(*

2.2.2.17 Pattern

A *pattern* entity belongs to a garment and defines the garment shape for a particular size. It is defined by a garment style name, a garment description for the operator to use, a set of *pattern_pieces*, and optionally, a tolerance value. Tolerances allow variations in measurements; tolerances are included only when variations are acceptable. If a tolerance value is defined in a *pattern* entity instance, the value is applied to all pattern pieces of the garment. However, the *basic_pattern_piece* allows this value to be redefined.

*)

ENTITY pattern;

style_name: STRING;

description: OPTIONAL STRING;

tolerance: OPTIONAL REAL;

pattern_pieces: SET [1:?] OF pattern_piece;

END_ENTITY;

(*

2.2.2.18 Grade Point

A *grade_point* entity specifies a grade point on a pattern piece. Points on a pattern piece may or may not be subject to a grading rule. A *grade_point* is defined by a unique *point* and optionally, an identifier of the grade point.

*)

ENTITY grade_point;

location: point;

identifier: OPTIONAL STRING;

UNIQUE location;

END_ENTITY;

(*

2.2.2.19 Grade Delta

A *grade_delta* entity specifies a displacement vector on a drawing. It consists of a *delta-X* value

and a *delta-Y* value. The *delta-X* and *delta-Y* values are the amounts of growth in the *X* and *Y* directions at the grade point between two grading sizes.

```
*)  
  
ENTITY grade_delta;  
  
    delta_x, delta_y: REAL;  
  
END_ENTITY;  
  
(*
```

2.2.2.20 Grade Rule At Point

A *grade_rule_at_point* is an abstraction of *grade_data_at_point* entity, and *library_rule_at_point* entity. Both subtypes inherit the grade point, the smooth option (which provides a means of determining that the grading curve will be generated with or without smoothing at the grade point), and the alternate grade reference line, if any. If an alternate grade reference line is defined in an entity instance, it overrides the grade reference line of the pattern piece that is given at the *grade_rules_of_piece* entity instance.

```
*)  
  
ENTITY grade_rule_at_point  
  
    ABSTRACT SUPERTYPE OF (  
  
        ONEOF (grade_data_at_point,  
  
                library_rule_at_point));  
  
    point: grade_point;  
  
    smooth_option: BOOLEAN;  
  
    alternate_grade_reference_line: OPTIONAL line;  
  
END_ENTITY;  
  
(*
```

2.2.2.21 Grade Data At Point

A *grade_data_at_point* entity, a subtype of *grade_rule_at_point*, is an ordered collection of the displacements, *grade_deltas*, of the specified grading point for a set of predefined grading sizes.

```
*)
```

```

ENTITY grade_data_at_point

    SUBTYPE OF (grade_rule_at_point);

    grade_deltas: LIST [1:?] OF grade_delta;

END_ENTITY;

(*)

```

2.2.2.22 Library Rule At Point

A *library_rule_at_point* entity, a subtype of *grade_rule_at_point*, specifies a grade rule identifier or label in the rule library that defines how an associated grade point on a pattern piece grows in the X and Y axes for each size in reference to the base size. The rule library is not represented in the schema. The entity is defined by a grade rule identifier. The *grade_rule_identifier* refers to an identifier for information that defines how an associated grade point on a pattern piece grows for each size in reference to the base size. The grade rule growth specifications are not represented in this schema.

```

*)

ENTITY library_rule_at_point

    SUBTYPE OF (grade_rule_at_point);

    grade_rule_identifier: STRING;

END_ENTITY;

(*)

```

2.2.2.23 Grade Rules Of Piece

A *grade_rules_of_piece* entity contains the information for grading a pattern piece for a set of predefined sizes. It consists of a set of *grade_rule_at_point*, and optionally, a rules identifier and a grade reference line. If a grade reference line is defined in an entity instance, it is applied to all grade points of the pattern piece.

```

*)

ENTITY grade_rules_of_piece;

    identifier: OPTIONAL STRING;

    grade_reference_line: OPTIONAL line;

    rules: SET [1:?] OF grade_rule_at_point;

```

```

        UNIQUE identifier;

    END_ENTITY;

    (*)

```

2.2.2.24 Grade Rules Of Pattern

A *grade_rules_of_pattern* entity contains the information for grading a pattern for a set of predefined sizes. It consists of an ordered collection of pattern sizes that include the base size, a set of *grade_rules_of_piece*, and optionally, a rules identifier.

```

    *)

    ENTITY grade_rules_of_pattern;

        identifier: OPTIONAL STRING;

        pattern_sizes: LIST [2:?] OF UNIQUE pattern_size;

        rules: SET [1:?] OF grade_rules_of_piece;

    END_ENTITY;

    (*)

```

2.2.2.25 Ready To Wear Pattern

A *ready_to_wear_pattern* entity defines graded pattern for all sizes. It is defined by a *pattern*, a *pattern_size* (the size of the base pattern), and optionally, a *grade_rules_of_pattern*. The *ready_to_wear_pattern* is defined with a common unit of measurement.

```

    *)

    ENTITY ready_to_wear_pattern;

        unit: measurement_unit_type;

        base_size: pattern_size;

        base_pattern: pattern;

        grade_rules: OPTIONAL grade_rules_of_pattern;

    END_ENTITY;

END_SCHEMA; -- end rwpm_schema

    (*)

```

3 EXPRESS-G FIGURES

In this section, the information model is presented by *EXPRESS-G*, a graphical subset of *EXPRESS*. *EXPRESS-G* is represented by graphic symbols forming a diagram. The notation has three types of symbols: definition (symbols denoting data types and schema declarations), relationship (symbols describing relationships which exist among the definitions), and composition (symbols enabling a diagram to be displayed on more than one page). *EXPRESS-G* is defined in Annex D of ISO 10303-11:1994(E), *EXPRESS* Language Reference Manual. Figures 3.1 through 3.10 correspond to the information model given in section 2.

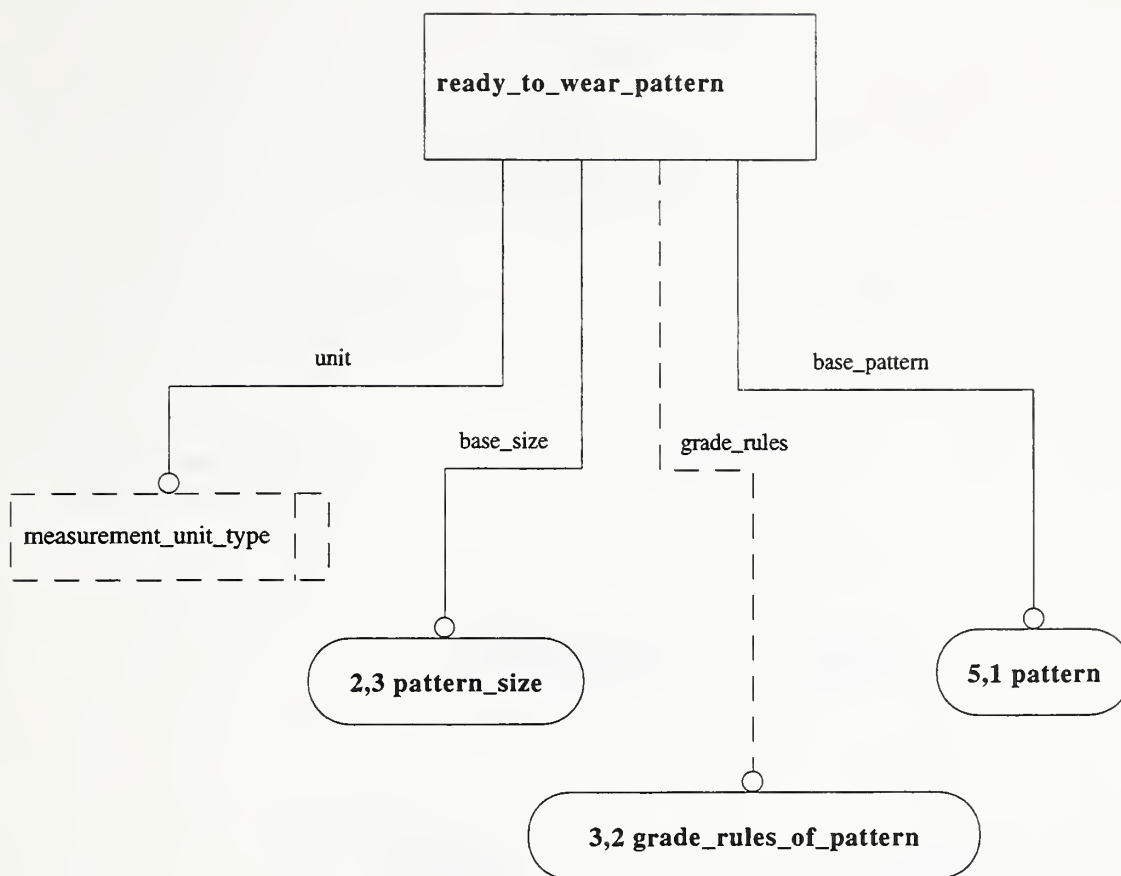


Figure 3.1 -- rwpm_schema EXPRESS-G diagram 1 of 10

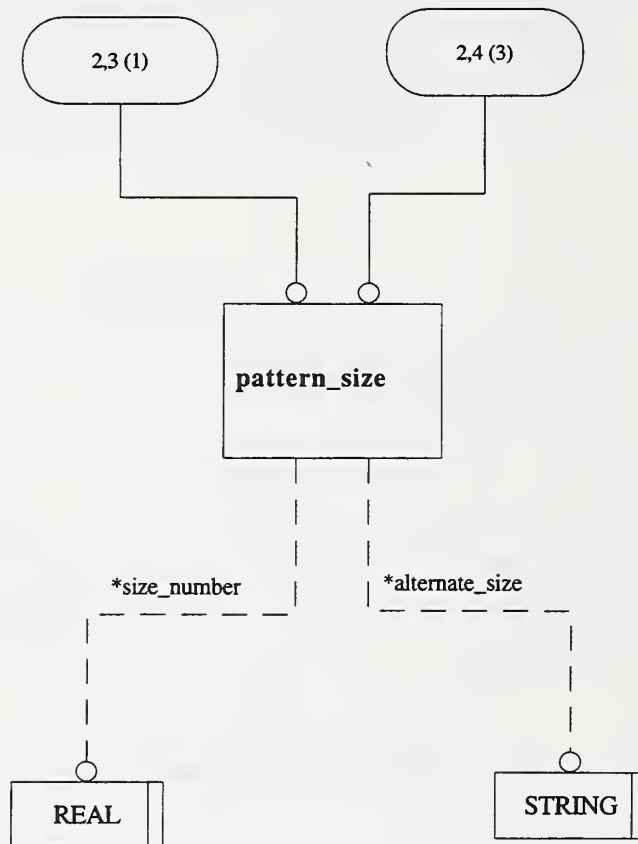


Figure 3.2 -- rwpm_schema EXPRESS-G diagram 2 of 10

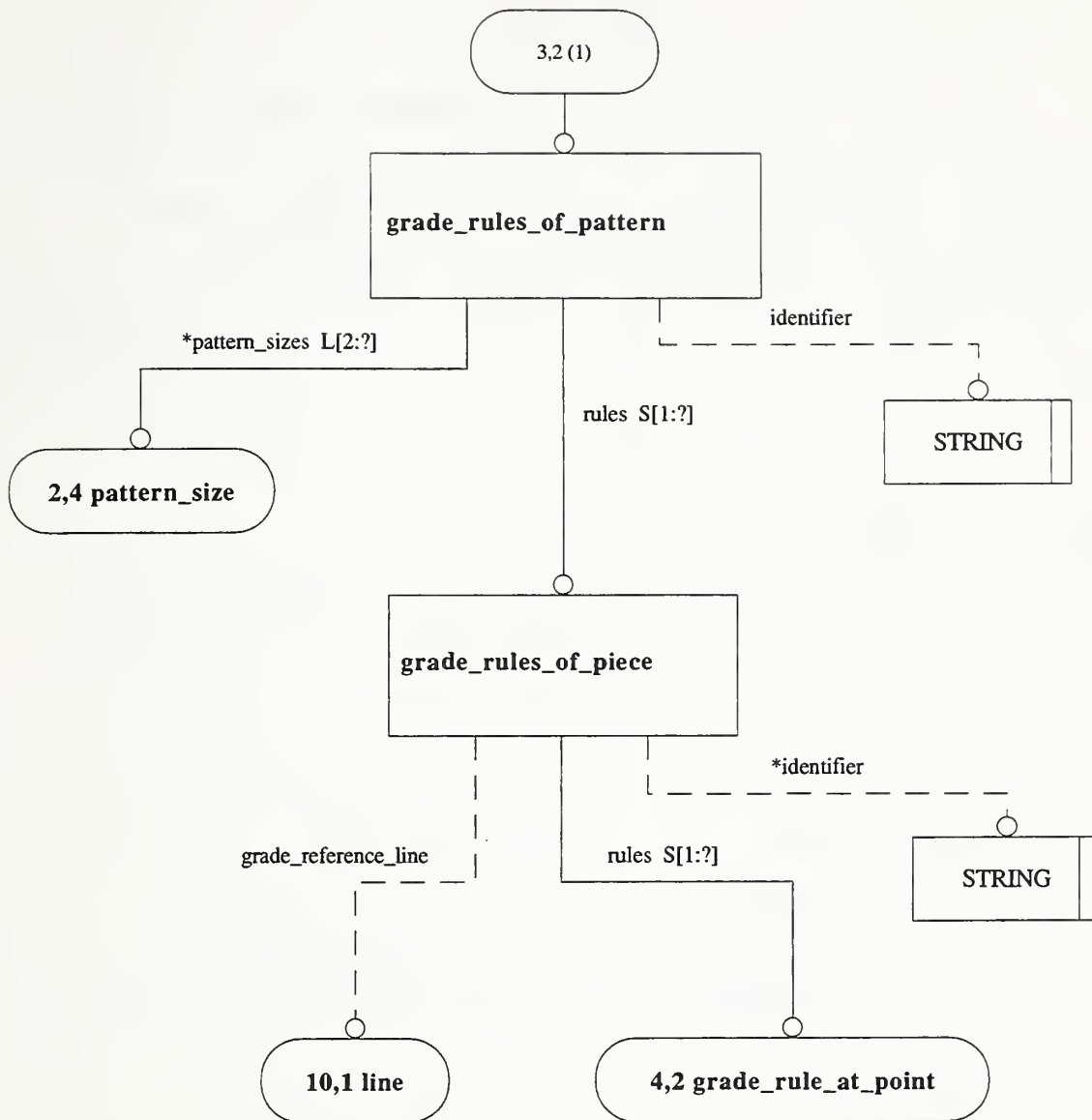


Figure 3.3 -- rwpm_schema EXPRESS-G diagram 3 of 10

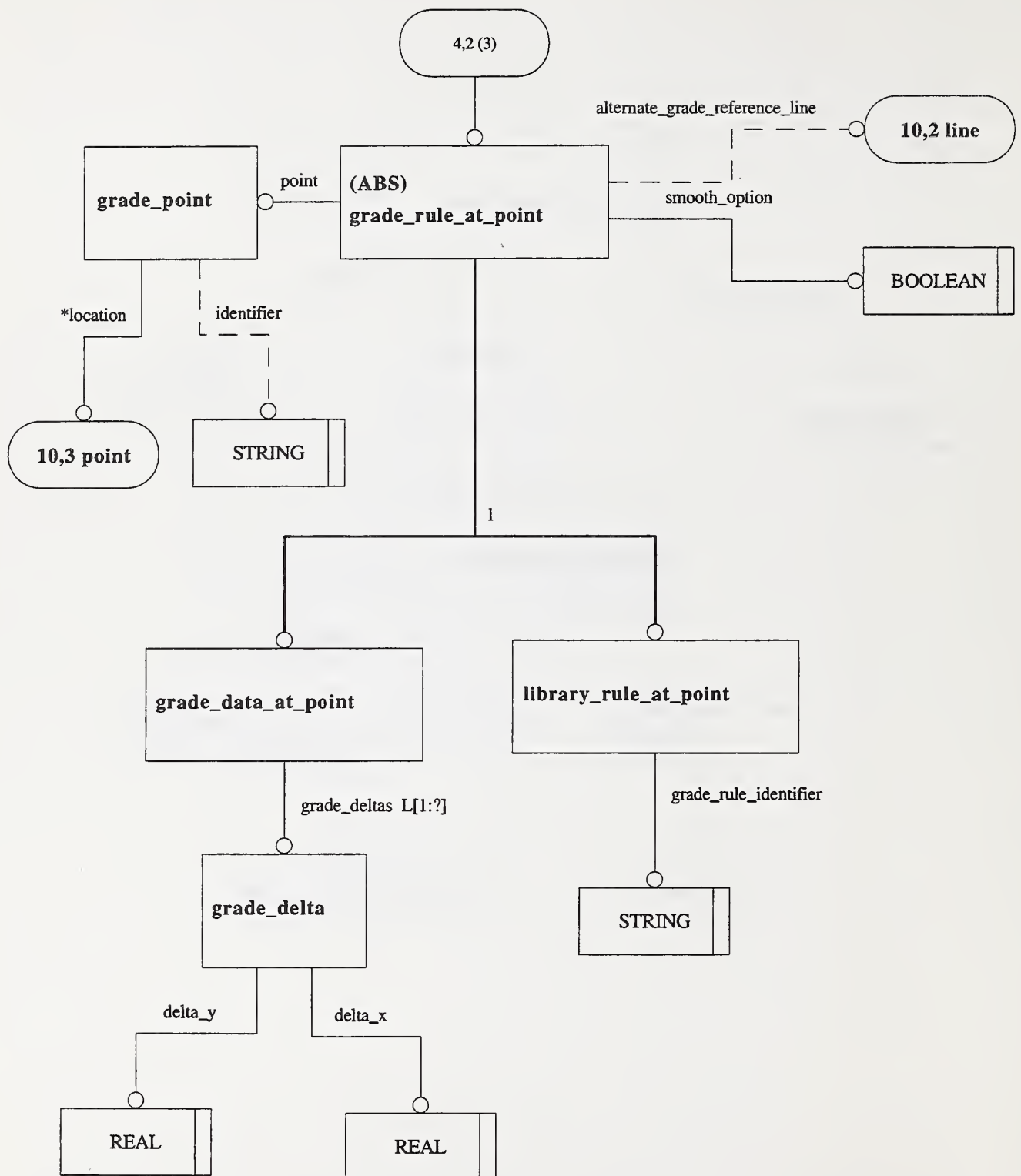


Figure 3.4 -- rwpm_schema EXPRESS-G diagram 4 of 10

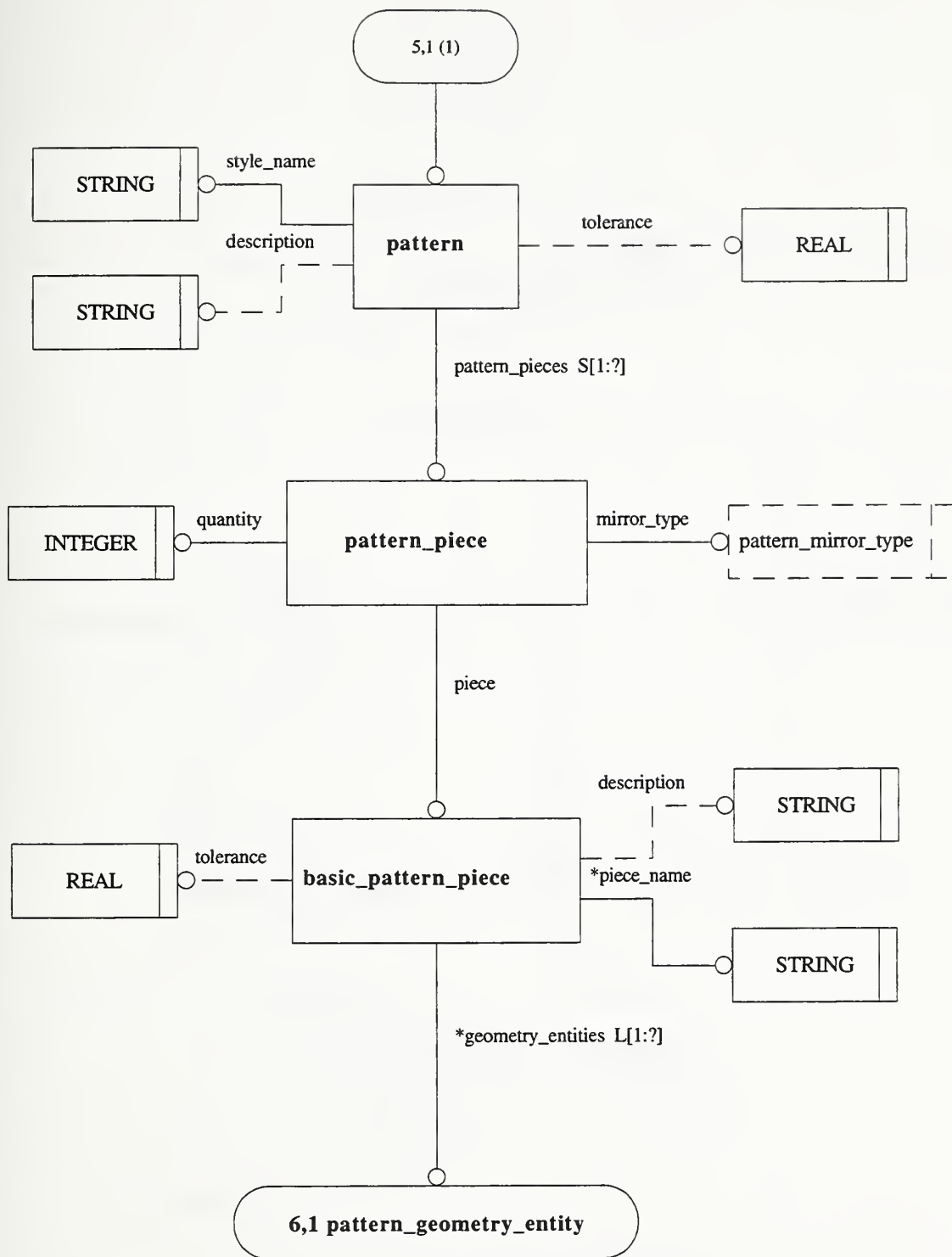


Figure 3.5 -- rwpm_schema EXPRESS-G diagram 5 of 10

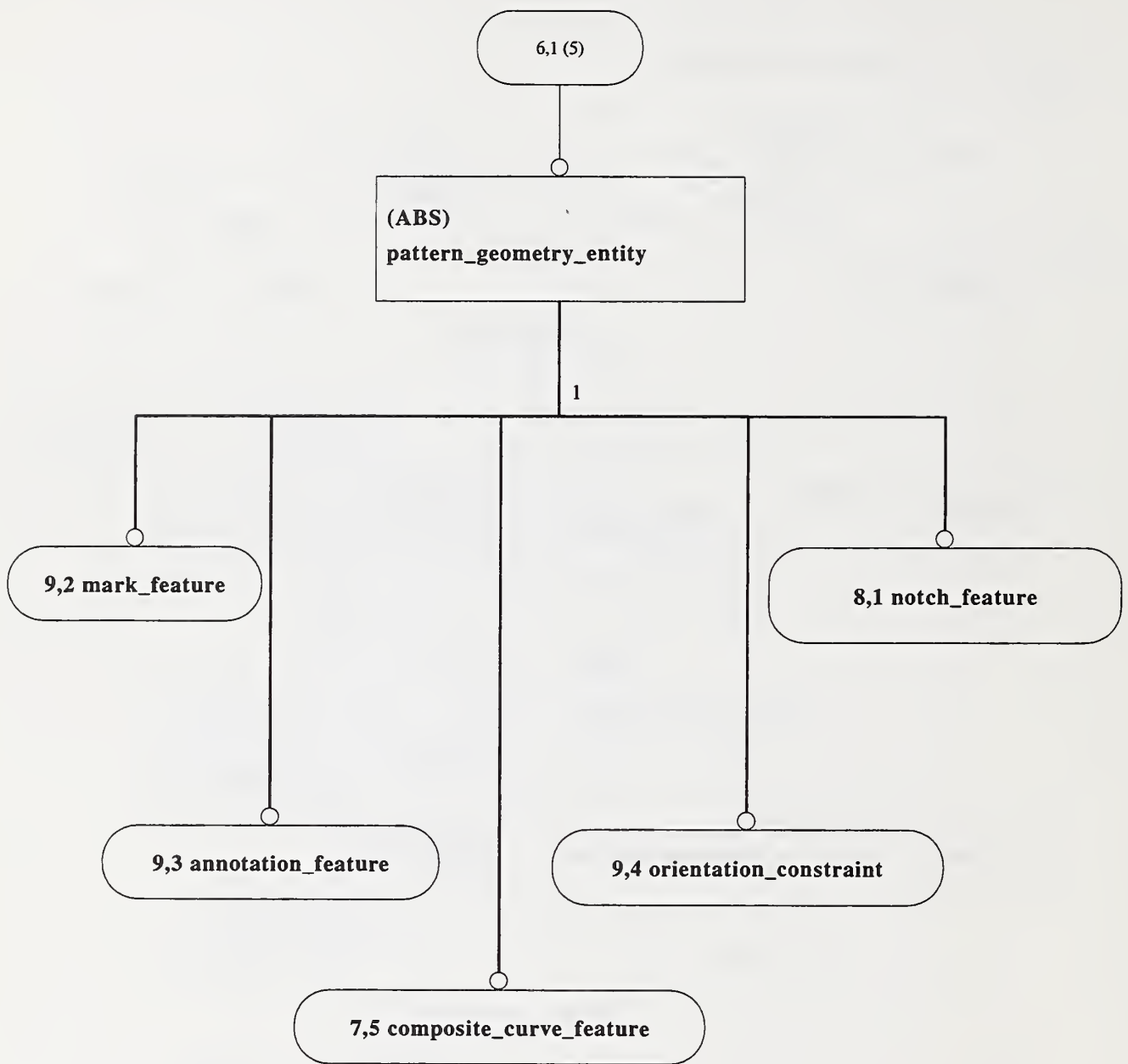


Figure 3.6 -- rwpn_schema EXPRESS-G diagram 6 of 10

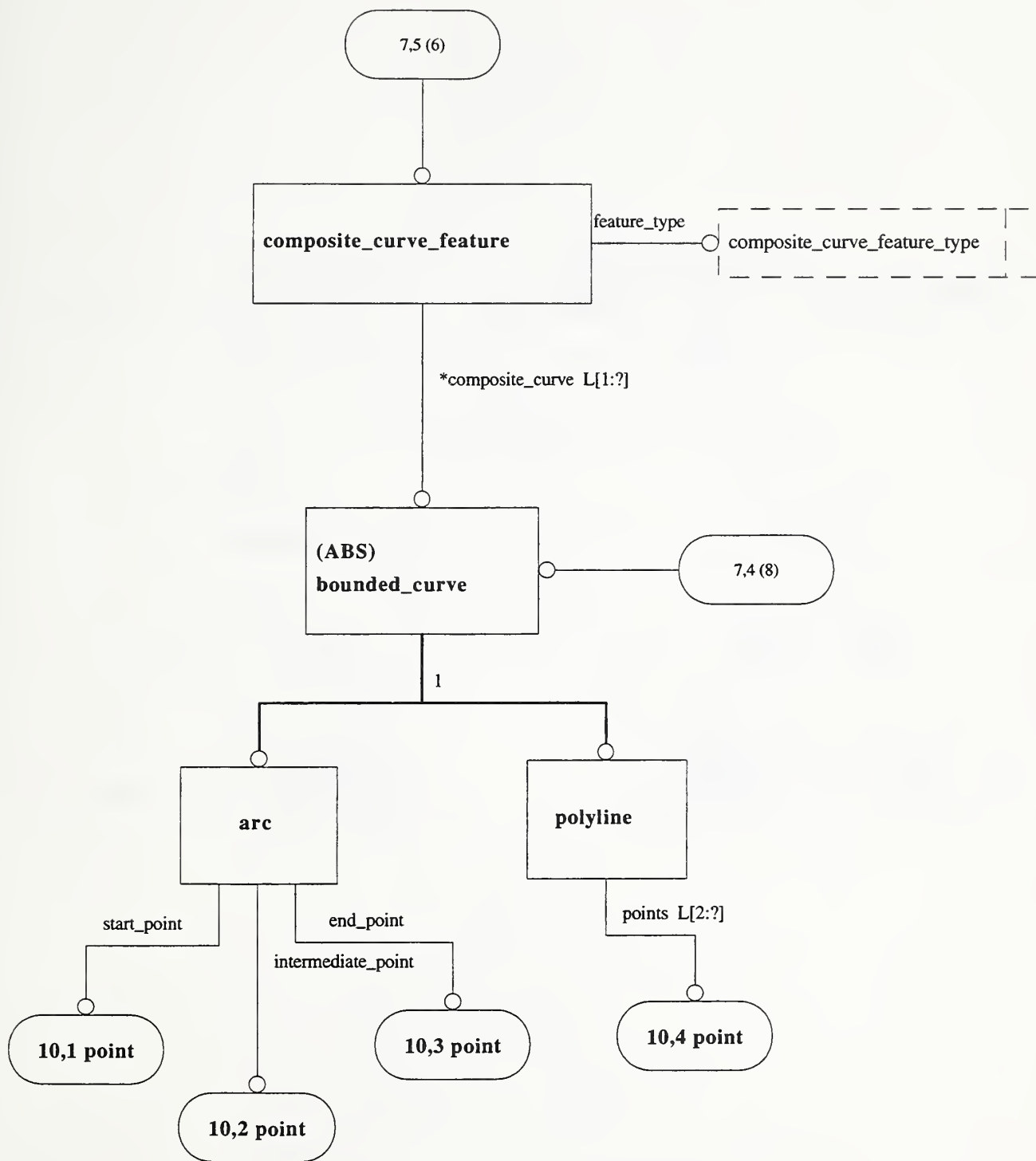


Figure 3.7 -- rwpm_schema EXPRESS-G diagram 7 of 10

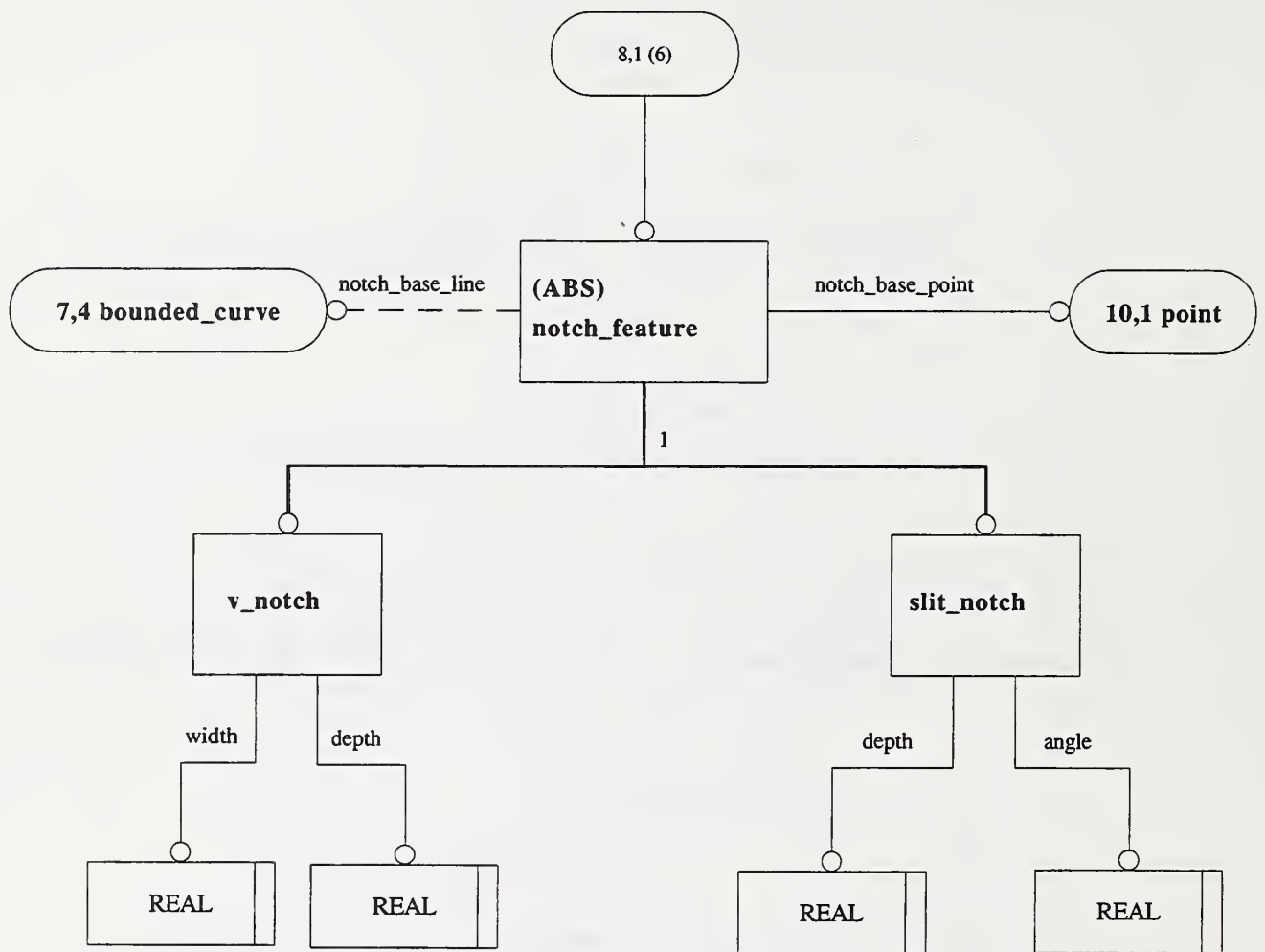


Figure 3.8 -- rwpn_schema EXPRESS-G diagram 8 of 10

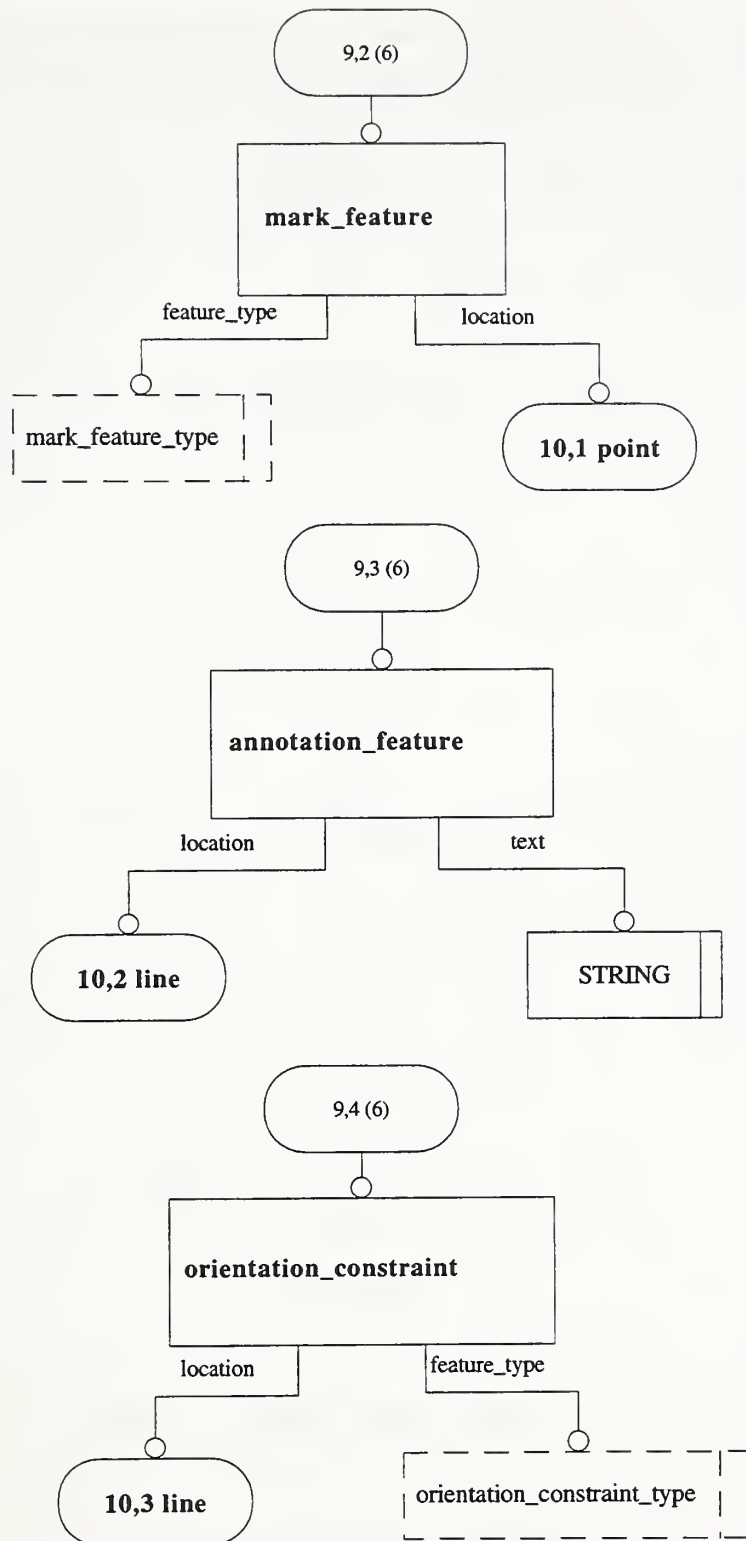


Figure 3.9 -- rwpm_schema EXPRESS-G diagram 9 of 10

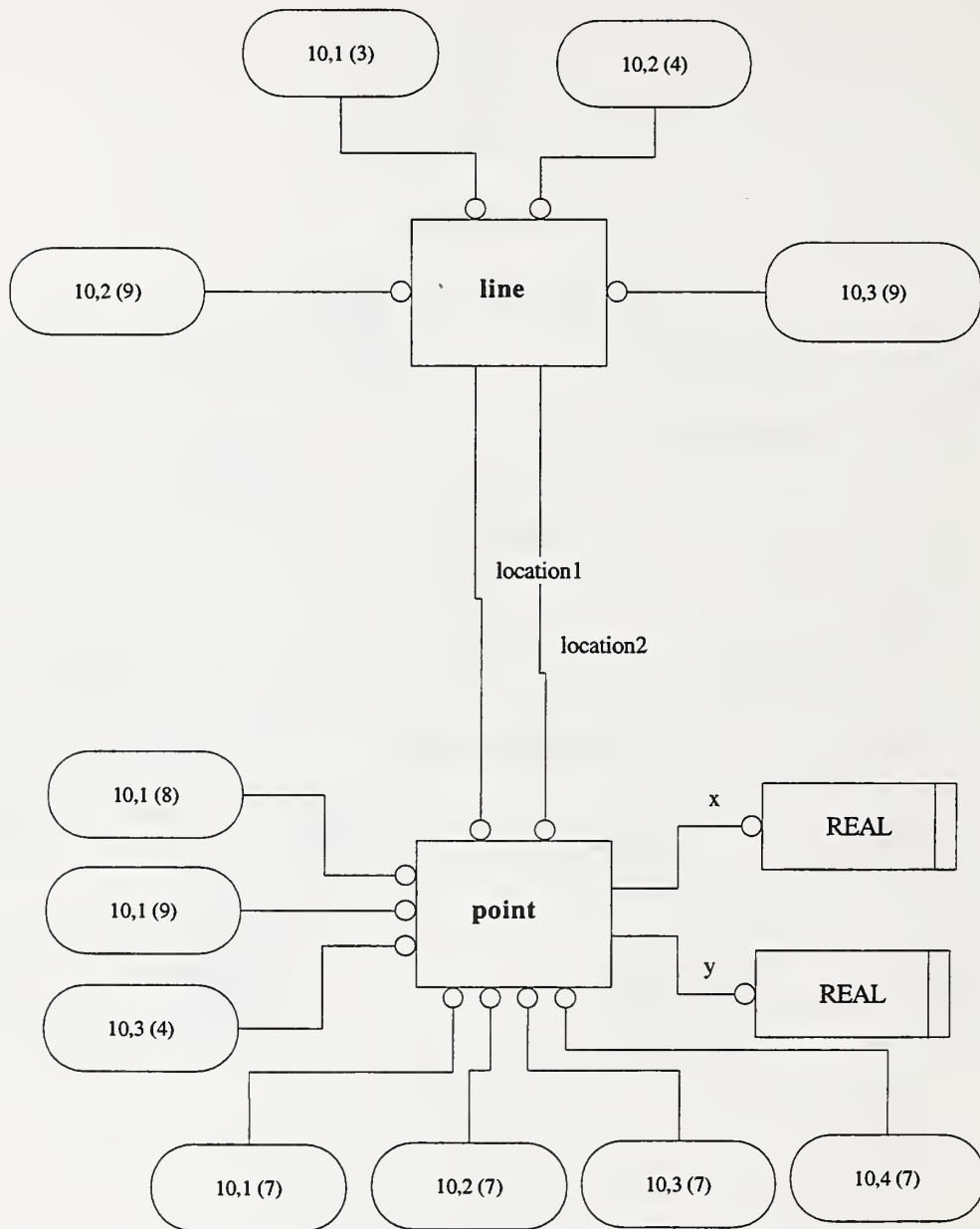


Figure 3.10 -- rwpm_schema EXPRESS-G diagram 10 of 10

4 APPLICATION ACTIVITY MODEL

The application activity model (AAM) is provided to aid in the understanding the scope and information requirements defined in the prototype application protocol of the ready-to-wear pattern making. The model is presented as a set of definitions of both the activities and the data and a set of activity diagrams. It covers activities which go beyond the subject of this application protocol. The definitions given in this section do not supersede the definitions given in the normative text. The diagrams use IDEF0 notation. Figure 4.1 shows the IDEF0 basic notation. Each activity may be decomposed to provide more detail. If an activity has been decomposed, a separate figure is included.

As with any IDEF0 model, the application activity model is independent on a particular viewpoint and purpose. The viewpoint of the application activity model is from a pattern maker or a marker maker. The purpose of the application activity model is to clarify the context and scope of the prototype application protocol.

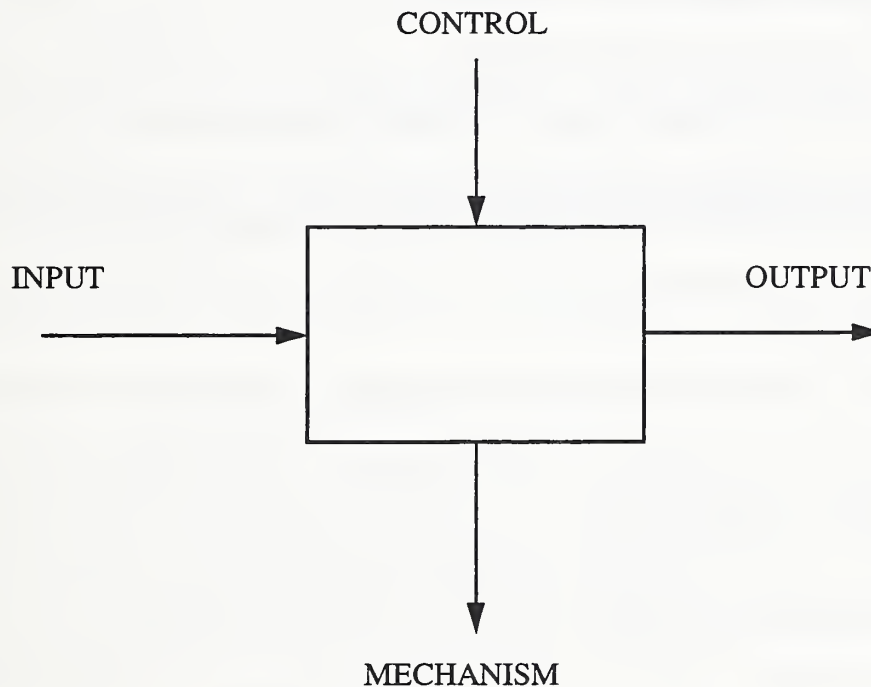


Figure 4.1 - IDEF0 basic notation

4.1 AAM Definitions

The following terms are used in the application activity diagrams. Terms marked with an asterisk are outside the scope of the application protocol.

BASE PATTERN - a pattern that is prepared for grading.

BASE PATTERN SPECIFICATION - information describing a collection of pattern pieces of a garment or style for a particular size.

BUSINESS COMMUNICATIONS* - information addressing all discussion and/or contact made with external organizations when operating an apparel manufacturer.

CREATE PROCESS PLAN* - the act of producing a sequence of construction operations involved in the manufacturing of a production order.

EQUIPMENT - a single machine or a group of related machines used to perform unit manufacturing operations.

FABRIC WIDTH* - data defining the width of the fabric.

GARMENT SIZES - information describing the fitting specifications of a garment.

GRADE PATTERN - the act of grading a sample size pattern to obtain pattern for a particular size.

GRADED PATTERNS - information describing individual patterns of a particular garment or style proportioned to a set of standardized body measurements for each size within a size range.

GRADING RULES - information describing rules for grading a sample size pattern to obtain patterns for other sizes.

MAKE BASE PATTERN - the act of laying out all pattern pieces of a garment or style for a sample size.

MAKE MARKER - the act of designing the guide for laying out the graded patterns into the marker section boundary to get the best fabric usage.

MARKER-MAKING SYSTEM - the equipment to produce a guide for pattern pieces placement on the fabric prior to spreading and cutting.

MARKER INFORMATION* - a guide for pattern pieces placement on the fabric prior to spreading and cutting.

MARKER PLAN* - plan to determine the marker making equipment and specify what sizes of garment will go into a marker in order to meet a production order.

MARKER SECTIONS* - information describing the location of scaled sections in a marker.

MATERIALS* - things needed for garment construction. This includes fabrics, findings, labels, etc.

PERSONNEL - individuals trained and employed by the organization.

PLAN FOR APPAREL PRODUCTION*- the act of creating a details of production plan, include schedule, process plan, material and resources arrangement, and spreading and cutting plan to fulfill the production orders.

POLICY & STANDARDS*- information addressing all government policies and standards, and apparel manufacturing industry standards that govern various aspects of apparel manufacturing enterprise's operating procedures.

PREPARE CUTTING - the act of grading patterns to various garment sizes for a production order and producing marker lay out.

PREPARE MATERIALS* - the act of determining the material requirements, ordering materials, receiving materials, and distributing materials in preparation for the production.

PREPARE RESOURCES* - the act of determining/assigning the equipments and personnel in preparation for the production.

PREPARED MATERIALS & INFORMATION* - all information relevant to materials that released to manufacturing.

PREPARED RESOURCES & INFORMATION* - all information relevant to equipments and personnel that involve in the production.

PRODUCTION ORDERS* - information about orders that requested by the customer and issued to the manufacturing plants for specific apparel products.

PRODUCTION PLAN* - information about the sequence of construction operations for the production.

RESOURCES* - information on equipments and human resources of the manufacturing plant.

SCHEDULE PRODUCTION* - determine production schedule for the manufacturing plant.

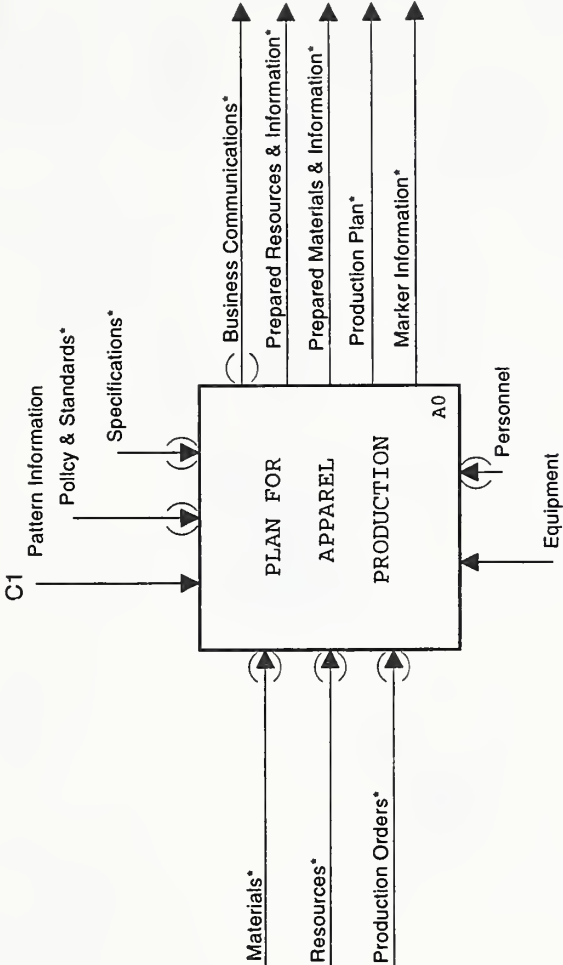
SPECIFICATIONS* - information addressing specific apparel products' requirements (such as materials, samples, construction), quality assurance, packaging, etc.

STYLE'S PATTERN INFORMATION - information containing base pattern and grading rules of a garment or style.

4.2 AAM Diagrams

The application activity model is given in the following three diagrams. The graphical form of the application activity model is presented in the IDEF0 activity modeling format. Activities and data flows which are out of scope are marked with asterisks.

USED AT:	AUTHOR: Tina Lee PROJECT: APDES NOTES: 1/ 2/ 3/ 4 5 6 7 8 9 10	DATE: 21 July 95 REV: 1.0		X	WORKING	READER	CONTEXT:
			DRAFT			DATE	
			RECOMMENDED				
			PUBLICATION				



- 1 Purpose: Clarify the context and scope of the Application Protocol for Ready-to-Wear pattern making.
- 2 Viewpoint: Pattern Makers and Marker Makers
- 3 Note: Activities and data flows that are out of scope of the AP are marked with asterisks.

NODE: A-0	TITLE: PLAN FOR APPAREL PRODUCTION (Context)	NUMBER: NIST-1
-----------	--	----------------

USED AT:	AUTHOR: Tina Lee PROJECT: APDES NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 21 July 95		WORKING <input checked="" type="checkbox"/>	READER 	DATE 	CONTEXT: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
		REV: 1.0					
		DRAFT					
		RECOMMENDED					
PUBLICATION							

C1

Pattern Information

MAKE
BASE
PATTERN
A51

Base Pattern Information →
 Grading Rules →

Base Pattern

Garment Sizes →

GRADE
PATTERN
A52

Graded Patterns →

Marker Sections*

MAKE
MARKER
A53

Marker Plan* →
 C3 →
 C2 →
 C4 →
 Fabric Width* →

Marker Information* → O1

MAKE

Marker-making System

NODE: A5	TITLE: PREPARE CUTTING	NUMBER: NIST-3
----------	------------------------	----------------

5 USE OF THE INFORMATION MODEL

The information model and the application activity model presented in this report can be used as the initial proposal for developing official STEP AP for apparel. An essential step is obtaining industry support and active participation towards a STEP standard. The process for developing an official AP, in full conformance with STEP standard, is described in "Guidelines for the development and approval of STEP application protocols, Version 1.1," prepared by the ISO standard subcommittee, TC184/SC4.

Several commercial and non-commercial software tools exist to support the development, implementation and validation of STEP APs. A document describing software tools and services for *EXPRESS* was published by P. Wilson and is available from the STEP On-Line Information Service (SOLIS) [7, 8]. NIST released a new version of the NIST STEP Toolset for manipulating STEP data in June 1995; the Toolset is in the public domain and is also available from SOLIS. The apparel industry can take advantage of these software tools to generate various type of data structures from the information model in order to benefit the exchange of pattern data between CAD systems.

REFERENCES:

- [1] Lee, Y. T., "Apparel Product Data Exchange Standard," Proceedings of the Third Annual Apparel Research Conference: Implementing Advanced Technology, Atlanta, GA, 1992.
- [2] Moncarz, H. T., and Lee, Y. T., "Report on Scoping the Apparel Manufacturing Enterprise," Vol. 5, No. 3/4, International Journal of Clothing Science and Technology, MCB University Press Limited, Bradford, UK, 1993.
- [3] ISO 10303-1:1994, "Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles," ISO, 1994.
- [4] Lee, Y. T., and Moncarz, H.T., "A Prototype Application Protocol for Ready-to-Wear Pattern Making," NISTIR 5115, National Institute of Standards and Technology, Gaithersburg, MD, 1993.
- [5] ISO 10303-11:1994(E), "Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 11: Description Methods: The EXPRESS Language Reference Manual," ISO, 1994.
- [6] Schenck, D., and Wilson, P., "Information Modeling the EXPRESS Way," Oxford University Press, New York /Oxford, 1994.
- [7] Wilson, P. R., "EXPRESS Tools and Services," EXPRESS Committee, 1994.
- [8] Rinaudot, G. R., "The IGES/PDES Organization: STEP On-Line Information Service (SOLIS)," NISTIR 5511, NIST, Gaithersburg, MD, 1994.

APPENDIX A:

The following is the list of *Express* keywords that are used in the RWPM model. Brief definitions of these keywords are presented for reader's convenience. Further information, refer to "EXPRESS Language Reference Manual".

ABSTRACT SUPERTYPE - An ABSTRACT SUPERTYPE is an entity type which possesses one or more subtypes and is not allowed to be directly instanced.

BAG - The key word BAG is used to specify a bag data type. A bag data type represents an unordered collections of like elements. The number of elements that can be held in a bag can optionally be specified. If the size is not specified, the bag can hold any number of elements. Duplicate elements are allowed in a bag.

BOOLEAN - A BOOLEAN data type represents a TRUE or FALSE value.

END_ENTITY - The key word END_ENTITY is used to terminate an entity declaration.

END_SCHEMA - The key word END_SCHEMA is used to terminate a schema declaration.

END_TYPE - The key word END_TYPE is used to terminate a type declaration.

ENTITY - The key word ENTITY is used to specify an entity type. An entity type characterizes a collection of real-world physical or conceptual objects which have common properties. Any entity declared in a schema can be used as the data type of an attribute, local variable or formal parameter. Using an entity as an attribute's data type establishes a relationship between the two entities.

ENUMERATION - The key word ENUMERATION is used to specify an enumeration data type. An enumeration data type is an ordered set of values represented by names. Each enumeration item belongs only to the data type which defines it and must be unique within that type definition.

EXISTS - The reserved word EXISTS is a function name. The EXISTS built-in unction is useful for checking if values have been given to OPTIONAL attributes, or if variables have been initialized. The EXISTS function returns TRUE if a value exists for the input parameter, or FALSE if no value exists for it.

INTEGER - The key word INTEGER is used to specify an integer data type. An integer data type represents a value of an integer number, the magnitude of which is unconstrained.

LIST - The key word LIST is used to specify a list data type. A list data type represents an ordered collections of like elements. The number of elements that can be held in a list can optionally be specified. If the size is not specified, the list can hold any number of elements. The Duplicate elements are allowed in a list.

OF - The key word OF is used together with other keywords such as BAG, LIST, SET, ENUMERATION, SUBTYPE, SUPERTYPE, etc.

ONEOF - The key word ONEOF is used to define the constraint on the relationship between the subtypes of a particular supertype. The ONEOF constraint is used when the subtypes are mutually exclusive.

OPTIONAL - The key word OPTIONAL is used to indicate that the attribute need not have a value in order for an instance of that entity to be valid. In a given entity instance, an attribute marked as optional may have no actual value, in which case the value is said to be null. The null value function (NVL) which returns either the input value or an alternate value in the case where the input has a null value may be used when a null value is unacceptable.

OR - The reserve word OR is an OR operator. The OR operator requires two logical operands and evaluates to a logical value.

REAL - The key word REAL is used to specify a real data type. A real data type represents rational, irrational, and scientific real numbers. Rational and irrational numbers have infinite resolution and are exact. Scientific numbers represent values which are known only to a specified precision.

SCHEMA - The key word SCHEMA is used to specify a schema type. A schema declaration creates a new scope in which the following objects may be declared: constant, entity, function, procedure, rule, and type.

SET - The key word SET is used to specify a set data type. A set data type represents an unordered collections of like elements. The number of elements that can be held in a set can optionally be specified. If the size is not specified, the set can hold any number of elements. No two elements of a set can have the same value.

STRING - The key word STRING is used to specify a string data type. A string data type represents a sequence of zero or more characters.

SUBTYPE - The key word SUBTYPE and SUPERTYPE are used for classification purposes. A subtype is a more specific type than its supertype(s). A supertype is a more general type than its subtype(s). Thus, every instance of a subtype is an instance of its supertype(s). An entity is a subtype if and only if it contains a non empty subtype clause. An entity does not have to declare itself to be a supertype. The subtype/supertype relationship is transitive. An entity is a supertype if it is named in the subtype clause of at least one other entity or declares itself to be an abstract supertype. A subtype may have more than one supertype, and a supertype may have more than one subtype. A supertype may itself be a subtype of one or more other entity types. Furthermore, a subtype cannot be the supertype of any type in the list of all its supertypes.

SUPERTYPE - See “subtype”.

TYPE - The key word TYPE is used to specify a defined data type. A defined data type is a user extension to the set of standard data types. A defined data type can be used as any other data type by referencing the name given to it.

UNIQUE - The key word UNIQUE is used to specify a unique rule. A unique rule specifies either

a single attribute name or a list of two or more attribute names. A rule which specifies a single attribute name is a “simple uniqueness constraint”, requiring that any value of that attribute is associated with only one instance of that entity type. A rule which specifies two or more attribute names is a “joint uniqueness constraint”, requiring that any set of values, one from each of the named attributes, is associated with only one instance of that entity type.

